IN THE CLAIMS

Claim Listing

1.   (currently amended) A computer implemented method for dynamically transforming and displaying, and optionally entering and updating data from XML data sources, said method comprising the steps of:

a.   reading a plurality of XML formats;

b.   defining a plurality of primary record types from components of the XML format that have multiple occurrences or have child components;

c.   defining a plurality of parent-child relationships between the primary record types from the relationships between the XML format components;

d.   defining a management record by choosing an XML format or by selecting a plurality of primary record types for inclusion in the management record;

e.   defining a dynamic document by choosing an XML format or by selecting a management record type, a plurality of its primary record types as-sociated with that management record type, and defining a hierarchical relationship for the selected primary record types;

f.   creating and updating primary record instances from XML documents;

g.   displaying data from the primary record instances using the dynamic document instances;

h.   creating an XML document definition for the dynamic document; and

i.   writing an XML document for the dymanic document instance.

2.   (original) The method of claim 1, wherein the step of defining a dynamic document includes any of the steps of:

a.   generating management record pointer families for each management record definition, each management record pointer family comprising a data structure defining a group of related primary record types within the management record, the management record pointer families generated by performing the steps of:

i.   retrieving a management record;

ii.   retrieving a unique primary record type of the management record;

iii.   using the unique primary record type as the current primary record type;

iv.   creating a family pointer type for the current primary record type;

v.   retrieving a set of primary record definitions related to the current primary record type;

vi.   creating a child record type to the family pointer for each primary record type retrieved in substep 5) that has a one to many relationship with the current primary record type:

and

vii.   for each primary record type retrieved in substep 5) that has a relationship with the current primary record type, repeating substeps 4-7 with the primary record type retrieved as the current primary record type;

b.   constructing the management record pointer instances;

c.   creating the dynamic document instances for the management record pointer instance.

3.   (currently amended) The method of claim 1, wherein the step of defining a primary record type further comprises the step of identifying a plurality of its fields, responsive to one of the XML formats, as the unique identifying fields for that primary record type.

4.   (original) The method of claim 1, wherein the step of defining a management record further comprises the step of identifying one of the selected primary record types as the unique identifying primary record for the management record.

5.   (original) The method of claim 1, wherein the step of defining a dynamic document includes any of the steps of:

a.   defining a function that uses the data of the data source;

b.   defining a filter that operates on the data of the data source, the filter specifying a set of data selection criteria; and

c.   defining a sort that operates on the data of the data source.

6.   (currently amended) The method of claim 1, wherein the step of creating and updating primary records from XML documents, further comprise the steps of:

a.   identify the primary record that corresponds to the XML format component;

b.   grouping into a temporary table the XML the component instances in response to by their key fields associated with those component instances into a temporary table; and

c.   inserting a primary record instance for each primary record type key field grouped by temporary table row.

7.   (original) The method of claim 1, further comprising the step of executing a function on the management records.

8.   (original) The method of claim 1, wherein the method executes user input sorts on the data in the dynamic documents before the step of displaying the data.

9.   (original) The method of claim 1, wherein the step of display-
ing data in the primary records using the dynamic document instance comprises
the steps of:

a.   sending the data in an organized format to a screen forms
display tool;

b.   placing values into proper fields; and

c.   using display rules entered by the user to control the appear-
ance of a set of display panes, positioning the format of fields within the primary
record type, and the type of maintenance that can be performed.

10.   (original) The method of claim 1, further comprising the step
of executing a filter on the dynamic document, the filter execution corresponding
to a data selection, the filter specifying a set of data selection criteria.

11.   (currently amended) The method of claim 10, wherein the
step of executing a filter on the dynamic documents further comprises the steps
of:

a.   reading the dynamic document;

b.   reading the filter from a dynamic document definition;

c.   comparing a filter value to an appropriate field of the dy-
namic document;

d.   displaying the dynamic document pointer if the filter value
matches the appropriate field of the dynamic document;

e.   determining whether the filter is a delete type filter; and

f.   deleting any dynamic document pointer and ~~its~~ children <u>as-</u>
<u>sociated therewith</u> if it is a delete type filter and the filter value matches the ap-
propriate field of the dynamic document.

12.   (original) The method of claim 1, further comprising the step
of executing a function on the management records.

13.   (original) The method of claim 12, wherein the step of exe-
cuting a function further comprises the steps of:

a.   retrieving the dynamic document definition and a dynamic
document instance;

b.   retrieving a lowest level unprocessed display pane hierarchy
level for the dynamic document definition;

c.   retrieving a set of functions associated with the display pane;

d.   executing each of the functions;

e.   updating the dynamic document; and

f.   updating a group by accumulator, wherein the group by ac-
cumulator stores a result corresponding to a calculation performed upon a group
of data fields.

14.   (original) The method of claim 2, further comprising the step
of executing a filter on the management records, the filter execution correspond-
ing to a data selection, the filter specifying a set of data selection criteria.

15.   (original) The method of claim 14, wherein the step of executing a filter further comprises the steps of

a.   reading the management record pointer instance;

b.   reading the filter from a management record definition;

c.   comparing a filter value to an appropriate field of the management record pointer instance;

d.   displaying the management record pointer instance if the filter value matches the appropriate field of the management record pointer instance;

e.   determining whether the filter is a delete type filter; and

f.   deleting any management record pointer instance and its children if it is a delete type filter and the filter value matches the appropriate field of the management record pointer instance.

16.   (original) The method of claim 2, wherein the step of executing a function further comprises the steps of:

a.   retrieving a management record definition and a management record instance;

b.   retrieving a lowest level unprocessed management record pointer family for the management record definition;

c.   retrieving a set of functions associated with the management record pointer family;

d.   executing each function retrieved;

e.   updating the management record pointer instances corre-

sponding to the management record pointer family;

f.   updating the parent management record pointer instances of

other management record pointer families that use the function; and

g.   (original) updating a group by accumulator, wherein the

group by accumulator Stores a result corresponding to a calculation performed

upon a group of data fields.

17.   (original) The method of claim 16, wherein the method

maintains a pass sequence number that is incremented for each level of function

nesting in a management record, and the step of executing the function is per-

formed a plurality of times equal in number to the pass sequence number.

18.   (original) The method of claim 16, wherein the method

maintains a pass sequence number that is incremented for each level of function

nesting in a dynamic document, and the step of executing the function is per-

formed a plurality of times equal in number to the pass sequence number.

19.   (original) The method of claim 2, wherein the step of defin-

ing a dynamic document comprises the step of defining dynamic document

pointers, said dynamic document pointers indicating the primary record types to

be displayed.

20.   (currently amended) The method of claim 19, the step of defining dynamic document pointers further comprises the steps of:

a.   retrieving a dynamic document definition;

b.   determining a highest hierarchy level display pane definition not yet processed, the display pane definition defining a display window;

c.   reading the primary record type and a corresponding management record pointer family for the pane definition;

d.   determining whether a display pane exists for the primary record type and the corresponding management record pointer family, the display pane corresponding to a display window; and

e.   if the display pane does not exist, creating a display pane for the primary record type and the corresponding management record pointer family by joining management record pointer instances for this display pane and a ~~its~~ parent display pane.

21.   (original) The method of claim 20, further comprising the step of executing filters before and after the step of creating a display pane.

22.   (original) The method of claim 20, further comprising the steps of:

a.   determining whether any child display panes exist; and

b.   storing a copy of the joining management record pointer instances for each child display pane.

23. (original) The method of claim 1, further comprising the steps of:

    a. accepting user revisions to the data displayed;

    b. determining whether the revision is a data revision;

    c. updating the management record instances if the revision is a data revision;

    d. determining whether the revision affects a function;

    e. maintaining and executing the affected functions if the revision affects a function;

    f. determining whether the revision affects a filter;

    g. maintaining and executing the affected filters if the revision affects a filter, each filter execution corresponding to a data selection according to a set of data selection criteria;

    h. determining whether the revision is a sort revision; and

    i. maintaining and executing the affected sort if the revision is a sort revision,

24. (original) The method of claim 23, wherein the step of maintaining and executing the affected functions if the revision affects a function comprises the steps of:

    a. retrieving user changes to the function;

    b. replacing calculated fields used in the function with their function;

    c.   updating primary record types used to point to the changed function;

    d.   updating the management record to point to the changed function;

    e.   reading the management record instance; and

    f.   executing the changed function.

    25.   (original) The method of claim 23, wherein the step of maintaining and executing the affected functions if the revision affects a function comprises the steps of:

    a.   retrieving user changes to the function;

    b.   replacing calculated fields used in the function with their function;

    c.   updating primary record types used to point to the changed function;

    d.   updating the dynamic document to point to the changed function;

    e.   reading the dynamic document; and

    f.   executing the changed function.

    26.   (original) The method of claim 23, wherein the step of maintaining and executing the affected filters if the revision affects a filter comprises the steps of:

    a.   retrieving user changes to the filter;

b.   updating the management record type definition to point to the changed filter;

c.   reading the management record instance for the filter's management record pointer family; and

d.   executing the filter.

27.   (original) The method of claim 23, wherein the step of maintaining and executing the affected filters if the revision affects a filter comprises the steps of:

a.   retrieving user changes to the filter;

b.   updating the dynamic document display pane definition to point to the changed filter, the display pane definition defining a display window;

c.   reading a display pane row for a display pane associated with the filter; and

d.   executing the filter.

28.   (original) The method of claim 23, wherein the display is updated after the step of maintaining and executing the affected sort if the revision is a sort revision.

29.   (original) The method of claim 23 wherein the step of updating the management record instances if the revision is a data revision, comprises the steps of:

a.   determining whether a record has been changed;

b.　updating the management record instances for a changed primary record instance if a record has been changed;

c.　determining whether an record has been added;

d.　updating the management record instances for a new primary record instance if a record has been added;

e.　executing the management record filters;

f.　executing the management record functions;

g.　updating the dynamic documents; and

h.　re-displaying the dynamic documents to show the data revision.

30.　(original) The method of claim 29, wherein the substep of updating the management record pointer instances for a changed primary record instance comprises the steps of:

a.　determining whether field values for a set of related fields are equal;

and b. removing a pointer within the management record pointer instance.

31.　(original) The method of claim 29, wherein the substep of updating the management record pointer instances for a new primary record instance comprises the steps of:

a.　retrieving the new primary record instance;

b.   determining where the new primary record instance is used by reading the primary record type definition and the management record type definition corresponding to the primary record instance;

c.   determining whether the primary record type is a lead primary record of a management record pointer family;

d.   creating a management record pointer instance and executing functions and filter associated with the management record pointer instance, if the primary record type is a lead primary record;

e.   if the primary record type is not a lead primary record, reading the management record pointer instance for the management record pointer family;

f.   determining whether a field related to the primary record instance is a management record pointer instance; and

g.   adding a primary record instance pointer to the management record pointer instance.

32.   (currently amended) The method of claim 29, wherein the step of executing the management record functions comprises the steps of:

a.   retrieving the changed primary record instance and a ~~its~~ corresponding primary record type;

b.   retrieving the function that uses the changed primary record instance;

c.   retrieving the management record pointer family of the function that uses the changed primary record instance;

d.   retrieving the management record pointer instance for the changed primary record instance;

e.   executing the function; and

f.   executing functions using the management record pointer instance.

33.   (original) The method of claim 29, wherein the substep of updating the dynamic documents comprises the steps of:

a.   identifying changed management record pointer instances and changed primary record instances;

b.   identifying the dynamic document pointers instances that use the changed fields;

c.   determining whether any of the changed fields, pointers and filters are used in other nested calculations, and identifying the dynamic document pointer instances that use the other nested calculations;

d.   executing the filters for the identified dynamic document pointers, each filter execution corresponding to a data selection according to a set of data selection criteria;

e.   executing the functions for the identified dynamic document pointers;

f.   re-sorting the records of dynamic document pointers; and

g.   re-displaying the records of dynamic document pointers.

34.   (currently amended) The method of claim 2, wherein the step of writing an XML document for the dynamic document instance when any ~~of its~~ dynamic document pointer records <u>associated with that dynamic document instance</u> change:

a.   identifying the changed primary record instances;

b.   identifying the dynamic documents pointer records that include changed primary record instances;

c.   updating the the dynamic documents pointer records affected by the primary record instance changes;

d.   retrieving the dynamic document definition to XML document type definition maps;

e.   creating the XML document by copying dynamic document instance data values into the record components;

f.   copying standard XML document header information and tags into the XML document; and

g.   handing the XML document and the destination Internet address to a web server.

35.   (currently amended) The method of claim 1, wherein

<u>a first user using a first dynamic document can view and maintain a first MRI,</u>

<u>a second user using a second dynamic document</u> ~~multiple individuals~~ can <u>concurrently</u> view and maintain <u>a second MRI,</u>

the <u>first MRI and the second MRI being substantially</u> identical ~~MRI~~

~~through multiple dynamic documents concurrently~~.

36.   (new) A method, including steps of

defining a plurality of primary record types responsive to a set of

components of an XML format that have multiple occurrences or have child

components;

defining an object representation in response to either an XML

format or a plurality of said primary record types;

defining a dynamic document in response to either an XML format

or in response to an object representation and a hierarchical relationship for a set

of selected primary record types; and

providing a technique for creating, displaying, or updating a set of

primary record instances in response to either XML documents or instances of

dynamic documents.

37.   (new) A method, including steps of

responsive to data in an XML format, defining a relational model

definition for said data;

defining a object model definition for said data, responsive to either

said XML format or said relational model; and

defining a mapping between said relational model and said object

model, said mapping being applicable to new data including an XML format;

whereby said new data is subject to inspection and manipulation using either a relational model or an object model.

38.   (new) A method as in claim 37, including steps of generating an XML format responsive to said new data.

39.   (new) A method as in claim 37, including steps of, responsive to said new data, generating a document including an XML format.

40.   (new) A method, including steps of

maintaining a set of data definitions and a set of relationship rules; and

responsive to a document in an XML format, defining a mapping between that XML format and said set of data definitions.

41.   (new) A method as in claim 40, including steps of manipulating that set of data definitions in response to data including an XML format.

42.   (new) A method as in claim 40, wherein those relationship rules include a relational database definition.

43.   (new) A method as in claim 40, wherein those relationship rules include

a relational database definition; and

at least one data relationship not substantially feasible using a relational database structure.

44. (new) A method as in claim 40, wherein those relationship rules include a substantially hierarchical structure of records.

45. (new) A method as in claim 40,

wherein those relationship rules include a substantially hierarchical structure of records; and

said records being responsive to relationships between a set of format components in that data including an XML format.

46. (new) A method as in claim 40, wherein those relationship rules include at least one of

a set of business rules;

a set of communication rules;

a set of document transaction rules.

47. (new) A method as in claim 40, including steps of generating a set of XML data definitions responsive to that document.

48. (new) A method as in claim 40, wherein that mapping is responsive to said relationship rules.

49.   (new) A method as in claim 48, wherein that mapping includes a relational database structure.

50.   (new) A method as in claim 48, wherein that mapping includes

a relational database structure; and

at least one technique for inspecting that data not substantially feasible using a relational database structure.

51.   (new) A method as in claim 40, including steps of

responsive to said mapping, defining a dynamic document;

said dynamic document including techniques for at least one of

importing that data responsive to the data definitions,

manipulating that data responsive to the relationship rules,

inspecting that data responsive to the relationship rules.

52.   (new) A method as in claim 40, including steps of

responsive to said mapping, defining a dynamic document; and

said dynamic document including techniques for generating a document including data from said dynamic document, formatted substantially in an XML format.

53.   (new) A method as in claim 40, including steps of

responsive to said mapping, defining a dynamic document; and

said dynamic document including techniques for generating a set of XML data definitions responsive to that dynamic document.

54. (new) A method as in claim 40, including steps of, responsive to actual data including an XML format, at least one of

importing that data responsive to the data definitions,

manipulating that data responsive to the relationship rules,

inspecting that data responsive to the relationship rules.

55. (new) A method as in claim 40, including steps of, responsive to actual data including an XML format, generating a document including data from said dynamic document, formatted substantially in an XML format.

56. (new) A method as in claim 40, including steps of, responsive to a second document in an XML format, said second document having a set of substantially distinct XML data definitions, including data from said second document in a unified database.

57. (new) A method as in claim 40, including steps of, responsive to a second document in an XML format, said second document having a set of substantially distinct XML data definitions, maintaining a unified instance of substantially duplicate XML components, whereby manipulating any one such instance affects all such instances.